Introduction to SQL

Class 4

Course Overview

- Introduction to SQL
  - Databases, Tables
  - Classification of SQL – DDL, DML, DCL, TCL
    - DDL – CREATE, ALTER, DROP
    - DML – SELECT, INSERT, UPDATE, DELETE
    - DCL – GRANT, REVOKE
    - TCL – COMMIT, ROLLBACK, SAVEPOINT
  - Data types, Operators
  - Keys – Primary, Foreign, Composite, Unique, Alternate
  - Integrity Constraints – Domain Integrity Constrains, Entity Integrity Constraints, Referential Integrity Constraints
  - Joins – Outer Joins, Left Outer Joins, Right Outer Joins, Inner Joins.
  - Queries, Subqueries, Functions, Flow Control (IF, CASE, WHILE, FOR, LOOP), Stored routines
  - Views
  - Concurrency and locking (Implicit locks, explicit locks, row level locks, table level locks, database level locks)
  - Indexes, Cursors, Triggers, Events
  - Tuning SQL queries and optimizing performance
  - SQL Databases vs NoSQL Databases
  - ACID, CAP
  - How SQL databases internally works

SELECT UNIQUE OR SELECT DISTINCT

- SELECT DISTINCT allergies,province_id FROM patients;

SELECT COUNT()

- Is a function that returns the number of records of the table in the output.
- Syntax
  - **SELECT** COUNT(column_name) **FROM** table_name;
- What is the difference between these two
  - Select count(allergies) from patients;
  - Select count() from patients;
  - Select count(*) from patients;
- COUNT(column_name) does not calculate NULL values
- COUNT can also be used with distinct
  - The DISTINCT keyword with the COUNT function shows only the numbers of unique rows of a column
  - **SELECT** COUNT(**DISTINCT** column_name) **FROM** table_name **WHERE** [condition];
  - But NULL is also a distinct value so it is also calculated

SELECT RANDOM

- RAND() function is used in MySQL which returns a random row
- It can be used in online exam to display the random questions.
- Syntax

```
SELECT column FROM table
ORDER BY RAND ( )
LIMIT 1
```

SELECT SUM

- It is a function used in a SQL query to return summed value of an expression.
- The expression can be a column name or a formula

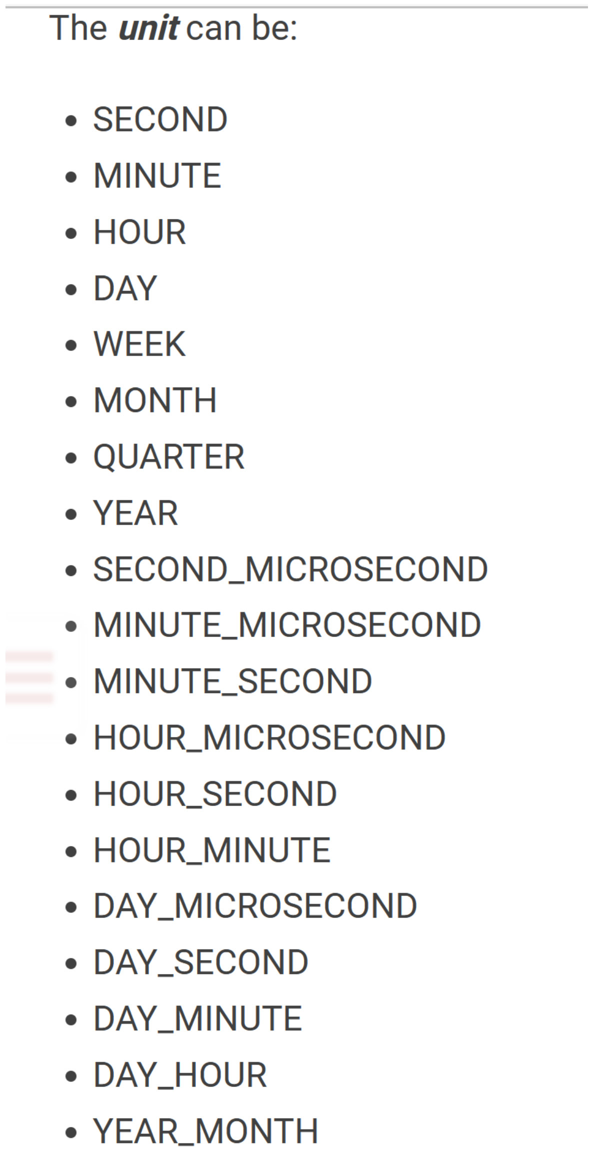
```
SELECT SUM (expression)
FROM tables
WHERE conditions;
```

Date functions

- SELECT CURDATE();
- SELECT DATE(datetime);
  - SELECT DATE('2021-01-17 10:12:16’);
- DATEDIFF(date1,date2);
  - SELECT DATEDIFF('2021-01-23','2021-01-14’);
- EXTRACT(unit FROM date);
  - SELECT EXTRACT(DAY FROM '2021-01-26’);
- MAKEDATE(year,day);
  - SELECT MAKEDATE(2021,34);
- HOUR(datetime);
  - SELECT HOUR('08:40:07');

DATE_ADD(date, INTERVAL value unit);

- SELECT DATE_ADD('2021-01-17 07:14:21', INTERVAL 20 MINUTE);

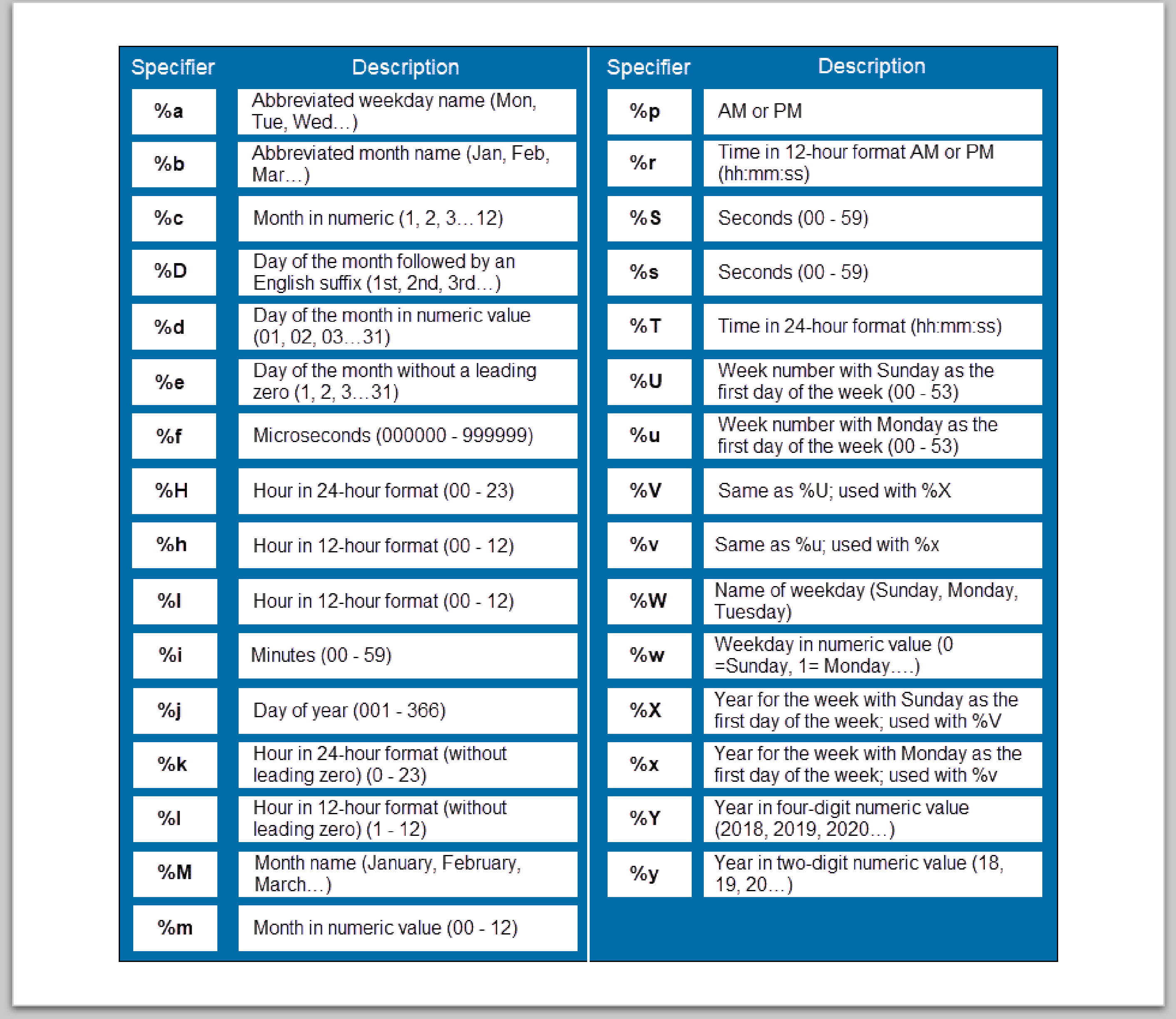


The ***unit*** can be:

- SECOND
- MINUTE
- HOUR
- DAY
- WEEK
- MONTH
- QUARTER
- YEAR
- SECOND_MICROSECOND
- MINUTE_MICROSECOND
- MINUTE_SECOND
- HOUR_MICROSECOND
- HOUR_SECOND
- HOUR_MINUTE
- DAY_MICROSECOND
- DAY_SECOND
- DAY_MINUTE
- DAY_HOUR
- YEAR_MONTH

DATE_FORMAT(date, format);

- SELECT DATE_FORMAT('2021-01-17', '%W %M %Y');



Specifier	Description	Specifier	Description
%a	Abbreviated weekday name (Mon, Tue, Wed…)	%p	AM or PM
%b	Abbreviated month name (Jan, Feb, Mar…)	%r	Time in 12-hour format AM or PM (hh:mm:ss)
%c	Month in numeric (1, 2, 3…12)	%S	Seconds (00 - 59)
%D	Day of the month followed by an English suffix (1st, 2nd, 3rd…)	%s	Seconds (00 - 59)
%d	Day of the month in numeric value (01, 02, 03…31)	%T	Time in 24-hour format (hh:mm:ss)
%e	Day of the month without a leading zero (1, 2, 3…31)	%U	Week number with Sunday as the first day of the week (00 - 53)
%f	Microseconds (000000 - 999999)	%u	Week number with Monday as the first day of the week (00 - 53)
%H	Hour in 24-hour format (00 - 23)	%V	Same as %U; used with %X
%h	Hour in 12-hour format (00 - 12)	%v	Same as %u; used with %x
%I	Hour in 12-hour format (00 - 12)	%W	Name of weekday (Sunday, Monday, Tuesday)
%i	Minutes (00 - 59)	%w	Weekday in numeric value (0 =Sunday, 1= Monday….)
%j	Day of year (001 - 366)	%X	Year for the week with Sunday as the first day of the week; used with %V
%k	Hour in 24-hour format (without leading zero) (0 - 23)	%x	Year for the week with Monday as the first day of the week; used with %v
%l	Hour in 12-hour format (without leading zero) (1 - 12)	%Y	Year in four-digit numeric value (2018, 2019, 2020…)
%M	Month name (January, February, March…)	%y	Year in two-digit numeric value (18, 19, 20…)
%m	Month in numeric value (00 - 12)		

String functions

- SELECT ASCII('2’); Returns numeric value of left most character
- SELECT CHAR_LENGTH("text"); Returns the length of the string
- SELECT CONCAT('My', 'S', 'QL’); Returns the concatenation of all inputs
- SELECT FORMAT(12332.123456, 4); Returns a rounded decimal number
- SELECT LENGTH('text’); Returns the length of the string
- SELECT LOWER('QUADRATICALLY’); Returns the lower case of the given string
- SELECT REPEAT('MySQL', 3); Returns the string repeated for the number of times

String functions

- REPLACE(str,from_str,to_str)
  - SELECT REPLACE('www.mysql.com', 'w', 'Ww’);
- REVERSE(str)
  - SELECT REVERSE('abcd’);
- SUBSTRING(str,pos)
- SUBSTRING(str FROM pos)
- SUBSTRING(str,pos,len)
- SUBSTRING(str FROM pos FOR len)
  - SELECT SUBSTRING('foobarbar' FROM 4);
- UCASE(str)

INSERT Statement

- Two ways to insert data
  - Insert a single row
  - Insert into multiple rows
- Inserting a single row where we have the column names specified

```
INSERT INTO table_name ( field1, field2,...fieldN )
VALUES
( value1, value2,...valueN );
```

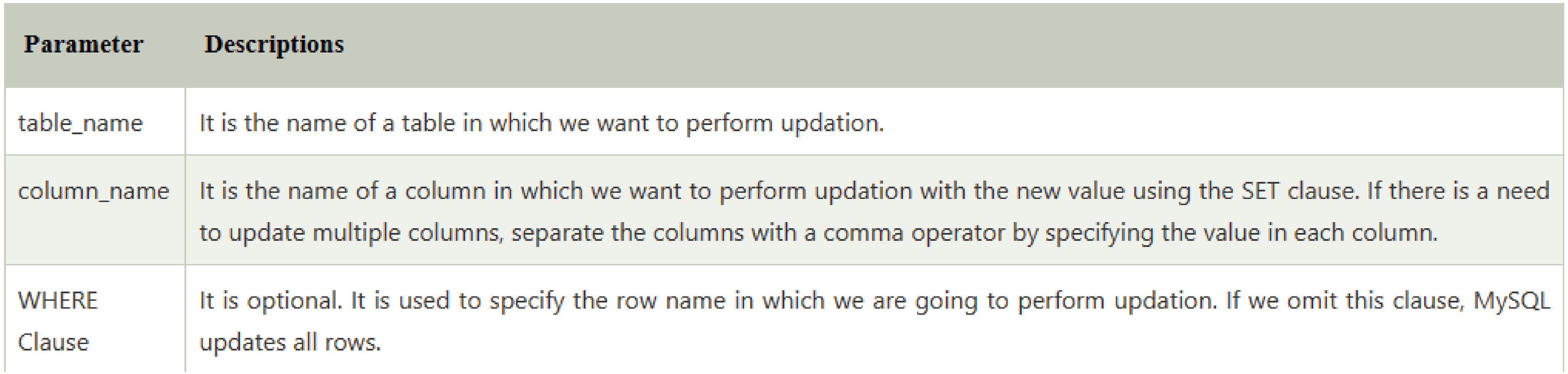
- Inserting multiple rows into a single command

```
INSERT INTO table_name VALUES
( value1, value2,...valueN )
( value1, value2,...valueN )
...........
( value1, value2,...valueN );
```

- We can also use SET command to insert data into a table
- We can also use a select statement to insert data into a table
- We need not have to specify column names if we are not omitting any columns while adding.
  - In that case we have to add values in such a way that they are in sync with how table was created i.e. we should not interchange columns

UPDATE Query

- Syntax
```
UPDATE table_name
SET column_name1 = new-value1,
      column_name2=new-value2, ...
[WHERE Clause]
```
- This statement can only update values in one single table at a time
- We can update single or multiple columns at a time
- Any condition can be specified in where clause
- If you miss a where clause, you will end up modifying everything. So careful
- We can also use a modifier keyword IGNORE –
  - Even if there are some errors while updating one column, we don’t have to stop other updates, they can continue. We use this keyword that time
```
UPDATE [LOW_PRIORITY] [IGNORE] table_name
   SET column_assignment_list
   [WHERE condition]
```



Parameter	Descriptions
table_name	It is the name of a table in which we want to perform updation.
column_name	It is the name of a column in which we want to perform updation with the new value using the SET clause. If there is a need to update multiple columns, separate the columns with a comma operator by specifying the value in each column.
WHERE Clause	It is optional. It is used to specify the row name in which we are going to perform updation. If we omit this clause, MySQL updates all rows.

DELETE Statement

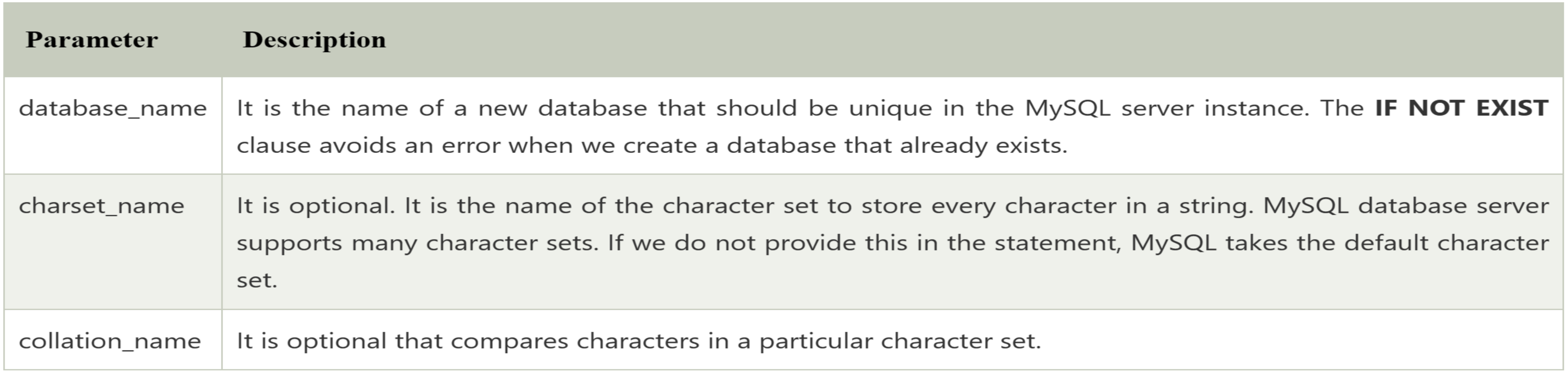
- It is used to remove rows that are no longer needed in the table
- This query can be used to delete one or more rows from a table
- We can also remove data based on condition
- But once deleted we cannot get the records back. They are gone forever. So it is better to keep a backup before we perform a delete
- Syntax
  - DELETE FROM table_name WHERE condition;
- If you forget the where clause, then all the rows will get deleted
- If we want to delete all the records without knowing the count of deleted rows, we use the TRUNCATE TABLE command.
  - This command gives better performance

Data definition languages

- CREATE
- ALTER
- DROP

DATABASE DDL Commands

```
CREATE DATABASE [IF NOT EXISTS] database_name
[CHARACTER SET charset_name]
[COLLATE collation_name];
```



Parameter	**Description**
database_name	It is the name of a new database that should be unique in the MySQL server instance. The **IF NOT EXIST** clause avoids an error when we create a database that already exists.
charset_name	It is optional. It is the name of the character set to store every character in a string. MySQL database server supports many character sets. If we do not provide this in the statement, MySQL takes the default character set.
collation_name	It is optional that compares characters in a particular character set.

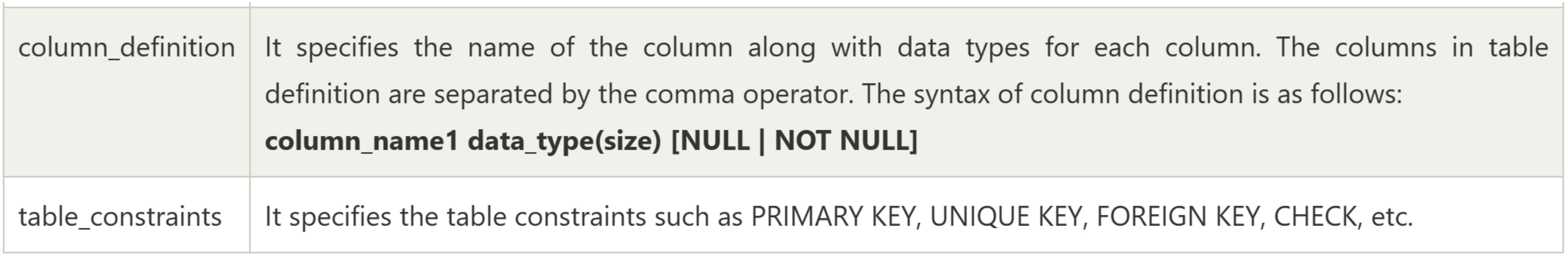
DATABASE DDL Commands

- USE database_name;
- SHOW DATABASES;
- **DROP DATABASE** [IF EXISTS] database_name;
    - We can also use the below statement because database and schema are one and the same
    - **DROP SCHEMA** [IF EXISTS] database_name;

CREATE TABLE

- Syntax

```
CREATE TABLE [IF NOT EXISTS] table_name(
    column_definition1,
    column_definition2,
    ........,
    table_constraints
);
```



column_definition	It specifies the name of the column along with data types for each column. The columns in table definition are separated by the comma operator. The syntax of column definition is as follows: **column_name1 data_type(size) [NULL | NOT NULL]**
table_constraints	It specifies the table constraints such as PRIMARY KEY, UNIQUE KEY, FOREIGN KEY, CHECK, etc.

Example

```
CREATE TABLE employee_table(
    id int NOT NULL AUTO_INCREMENT,
    name varchar(45) NOT NULL,
    occupation varchar(35) NOT NULL,
    age int NOT NULL,
    PRIMARY KEY (id)
);
```

Connect with me on LinkedIn

Please subscribe to our YouTube channel

Check out my GitHub profile

Follow me on Twitter(X)

Thank you